



Docket No.: F1866.0065

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Tomohiro Igakura

Application No.: 09/960,548

Confirmation No.: 5904

Filed: September 20, 2001

Art Unit: 2161

For: FILE MANAGING SYSTEM

Examiner: T. Y. Chen

RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is a response to the Notice of Non-Compliant Appeal Brief dated December 21, 2006. Please substitute the attached Appeal Brief for the Appeal Brief previously submitted in the above-identified matter.

As required under § 41.37(a), this brief was originally filed within two months of the Notice of Appeal filed in this case on May 27, 2005, and is in furtherance of said Notice of Appeal.

No fee is believed to be due with this submission. However, in the event a fee is required or if any additional fee during the prosecution of this application is not paid, the Patent Office is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 50-2215.

CONTINGENT EXTENSION REQUEST

If this communication is filed after the shortened statutory time period had elapsed and no separate Petition is enclosed, the Commissioner of Patents and

Trademarks is petitioned, under 37 CFR 1.136(a), to extend the time for filing a response to the outstanding Office Action by the number of months which will avoid abandonment under 37 CFR 1.135. The fee under 37 CFR 1.17 should be charged to our Deposit Account No. 50-2215.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

- I. Real Party In Interest
- II Related Appeals and Interferences
- III. Status of Claims
- IV. Status of Amendments
- V. Summary of Claimed Subject Matter
- VI. Grounds of Rejection to be Reviewed on Appeal
- VII. Argument
- VIII. Claims
- IX. Evidence
- X. Related Proceedings
- Appendix A Claims

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

NEC Corp.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 17 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-17
4. Claims allowed: None
5. Claims rejected: 1-17

C. Claims On Appeal

The claims on appeal are claims 1-17

IV. STATUS OF AMENDMENTS

Appellant filed an Amendment After Final Rejection on March 31, 2005. The Examiner contacted the Appellant's representative to discuss the arguments presented in the March 31 Response. Appellant did not have the file and was unable to discuss the claims in depth. Appellant's representative called the Examiner back but the Advisory Action of May 4, 2005, was already mailed. In the Advisory Action, the Examiner indicated that Appellant's proposed amendments to the claims would not be entered. However, there were no unentered amendments at the time the Advisory Action was issued. Further, as discussed below in Section VII, Appellant disagrees with the Examiner's assumption that Appellant agrees with the Examiner's mapping. See Advisory Action at page 3.

Accordingly, the claims enclosed herein as Appendix A incorporate the amendments filed by Appellant on August 27, 2004, considered in the Final Office Action mailed March 31, 2005.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claims 1-17 are pending in this application. Of those claims, 1-3, 7-9 and 12 are independent. Appellants will address each of the limitations in the independent claims the first time such a limitation appears in each claim.

A. Claim 1

Independent claim 1 explicitly recites:

1. A file managing system for managing files comprising:
a manager for managing a plurality of files having the same file title but different contents as separate files, and for managing a plurality of files having the same content but different titles as a single file.

The present invention relates to a file management system. Particularly, the claimed filed system is able to manage files having the same title but different contents as separate files (See p. 1, lines 6 – 13) while managing a plurality of files having different titles but the same content as a single file. See p. 30, lines 26 – 27. For example, three files can be named “file A.” Assume that these three files have different contents – specifically function A, function B, and function C. Because each of these files have different contents, the files will have different Managing IDs. See p. 16, lines 19 – 25. Had two or more of these files had the same contents, the files with the same contents would share a common Managing ID. See p. 15, lines 15 – 25.

B. Claim 2

Independent claim 2 explicitly recites:

2. A file managing system for managing files, comprising:
a file input unit for sending, to a data processor, inputted files, file titles, and file IDs to be registered;

a file request input unit for sending, to the data processor, a file title of a requested file and a file ID;

a memory unit including a correspondence table and a file memory, the correspondence table including correspondence relationships of file titles, file IDs and managing IDs, the file memory for recording, managing IDs and files;

the data processor including:

a means for producing and recording, if no file has the same content as any of the files recorded in the file memory, a new managing ID for a new file to be registered in the file memory;

a means for retrieving files from the file memory using managing IDs;

a means for sending the retrieved file to a file outputting unit,

a file content comparing means for comparing the content of the new file to be registered and the contents of files registered in the file memory,

a means for registering, if a same content file has been registered in the file memory, the file title, the file ID, and the managing ID of the same content file in the correspondence table, and

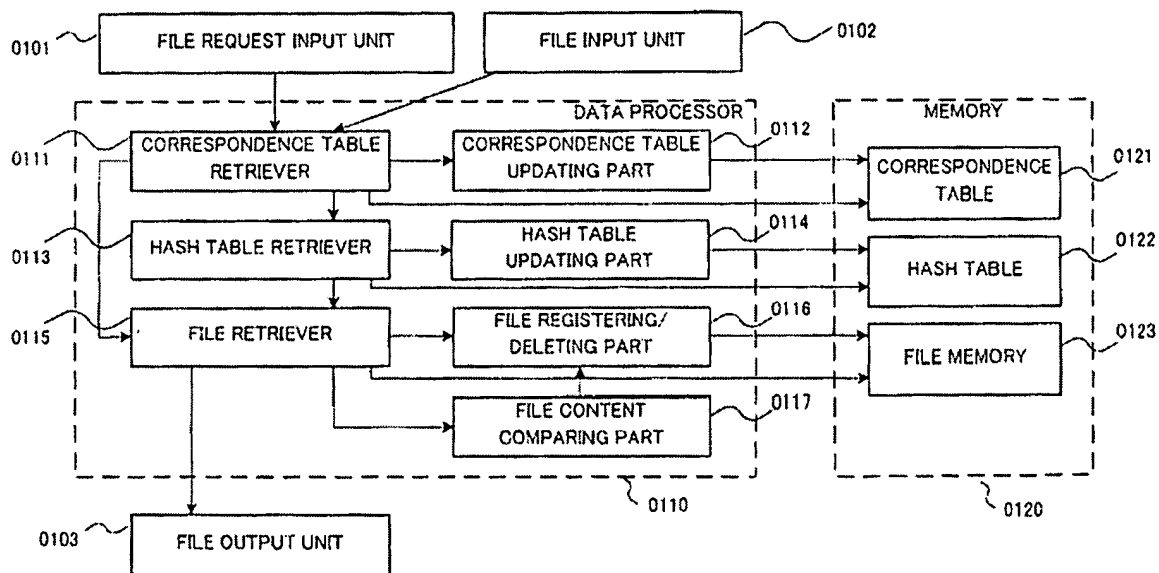
a means for retrieving the correspondence table; and

the file output unit for sending, to the file request input unit, the file corresponding to the file title and the file ID requested from the file request input unit as delivered from the data processor.

The present invention relates to a file management system. Particularly, the claimed filed system is able to manage files having the same title but different contents as separate files while managing a plurality of files having different titles but the same content as a single file. See p. 1, lines 6 – 13; p. 30, lines 26 – 27.

According to the present invention, when a file ID and a file are input into the system using file input 102, the system first checks whether the title was previously

registered using “means for registering”. See p. 20, lines 2 – 10. As shown in Fig. 1 reproduced below, a file, file title, and file ID are input into the system using file input 102. The information is then transferred to data processor 110 when the above-described process takes place.



When requesting a file, a file requestor inputs a file title and file ID using file request input unit 101. The correspondence table retriever 0111 then retrieves the correspondence table 0121 for any file having the same file title as the file to be newly registered. See Fig. 5, step 0502; pg. 20, lines 2-5. It is inherent that the file title is input because, any file having the same title is retrieved because if no title is entered the system would not know what files to retrieve.

The system then verifies the correspondence table 121 with a key with a given file name and file ID. See Fig. 6, 602. The correspondence table is retrieved using correspondence table retriever 111. See Fig. 1. If a managing ID is retrieved from the correspondence table, the file is retrieved from memory 120 that includes correspondence table 121 and file memory 123 and output to the user. If no managing

ID is obtained from the correspondence table, an error is returned to the user. See page 22, lines 10 – 26. The file is retrieved by “a means for retrieving files” which corresponds to at least file retriever 115.

New managing IDs are produced and recorded, if required, by data processor 110. Data processor 110 includes correspondence table retriever 111, correspondence table updating part 112, hash table retriever 113, hash table updating part 114, file retriever 115, file registering/deleting part 116, and file content comparing part 117. Specifically, the “means for producing and recording ... a new managing ID” includes at least file retriever 0115 which provides a function of producing a new managing ID for a file to be registered and file registering/deleting part 0116 registers the given file in the file memory 0123. See, page 18, lines 7-15.

The file is retrieved by “a means for retrieving files” which corresponds to at least file retriever 115. The file retriever 0115 also sends a retrieved file to the file output unit 0103. Thus, the file retriever 115 also corresponds to the means for sending the retrieved file to a file outputting unit 103.

File content comparing part 0117, which generally corresponds to “file content comparing means,” compares the content of a file to be registered and the content of the files registered in the file memory 0123. If a managing ID is obtained as a result of the hash table reference, the file to be registered is compared to the file from the hash table. If the content is the same, the file management ID is set so that it corresponds to the file to be registered. In contrast, if the content is different, the managing ID is registered in a hash table 122. See page 28, lines 3 – 20; Fig. 7.

If the file name already exists, the system checks a hash table 122 to determine whether or not the file has been previously registered. See p. 20, line 11 – 14; Fig. 5. If there is no identical hash value, which indicates identical content, the file has not been previously registered. If the hash value exists – the file to be registered is compared with the files corresponding to the managing ID obtained from the hash table

122. See Fig. 5, 506. If it is determined that a file has the same content as a previously registered file, its file management ID is set to correspond to the registered file. However, if there is no file having the same content, a new managing ID corresponding to the file is created and registered. See Fig. 5, 508; page 14, line 8-page 16, line 18, page 21, lines 13 – 22. The file is then stored with the managing ID and a correspondence table 121 is supplied by the correspondence table updating part 112 of processor 110. The managing ID corresponding to the file and file ID are finally registered for retrieval. See page 22, lines 1 – 5 and Fig. 5.

If the file name already exists, the system checks a hash table 122 to determine whether or not the file has been previously registered. See p. 20, line 11 – 14; Fig. 5. If there is no identical hash value, which indicates identical content, the file has not been previously registered. If the hash value exists – the file to be registered is compared with the files corresponding to the managing ID obtained from the hash table 122. See Fig. 5, 506. If it is determined that a file has the same content as a previously registered file, its file management ID is set to correspond to the registered file. However, if there is no file having the same content, a new managing ID corresponding to the file is created and registered. See Fig. 5, 508; page 14, line 8-page 16, line 18, page 21, lines 13 – 22. The file is then stored with the managing ID and a correspondence table 121 is supplied by the correspondence table updating part 112 of processor 110. The managing ID corresponding to the file and file ID are finally registered for retrieval. See page 22, lines 1 – 5 and Fig. 5. Data processor 110 includes correspondence table retriever 111. Finally, file output unit 103 sends the file 2 as delivered from the data processor 110.

C. Claim 3

Independent claim 3 explicitly recites:

3. A file managing system for managing files, comprising:

- a file input unit for sending, inputted files file titles, and file IDs to be registered to a data processor;

- a file request input unit for sending to the data processor, an inputted file title and a pertinent file ID;

- a memory unit including;

- a correspondence table, in which correspondence relationships of file titles, file IDs and managing IDs are recorded;

- and a file memory, in which managing IDs and files are recorded;

- a data processor including:

- a means for producing a new managing ID and registering, in the file memory, the new managing ID and a file to be registered,

- a file deleting means for deleting, if a same content file has been registered in the file memory, the managing ID and the file registered in the file registering means,

- a means for retrieving the file memory with managing IDs for obtaining corresponding files,

- a means for sending out the obtained files to a file output unit,

- a file content comparing means for comparing the content of a file to be registered with the contents of the files registered in the file memory,

- a correspondence table registering means for registering, in the correspondence table, the file titles to be registered, the file IDs to be registered and the new managing IDs,

- a correspondence table updating means for updating, if a same content file has been registered in the file memory, the contents registered in the correspondence table registering means with the file titles to be registered, the file IDs to be registered and the managing ID of the same content file, and

a means for retrieving the correspondence table; and
the file output unit sending out, to the file request input unit,
the file corresponding to the file title and the file ID requested
from the file request input unit as delivered from the data
processor.

The present invention relates to a file management system. The limitations
not addressed here are addressed above.

Claim 3 recites file deleting means. If a managing ID is obtained as a result
of the hash table reference, the file to be registered is compared to the file from the
hash table. If the content is the same, the file management ID is set so that it
corresponds to the file to be registered. In contrast, if the content is different, the
managing ID is registered in a hash table 122. See page 28, lines 3 – 20; Fig. 7. When
the same content file has been registered in the file memory, the hash table updating
unit 114 of deletes the given managing ID from the hash table. Thus, hash table
updating unit 114 corresponds to the claimed “file deleting means.” Additionally, the
“file deleting means” can be interpreted as the file registering/deleting part 0116 which
deletes the files registered in the file memory 0123 together with the managing IDs of
these files. See page 28, lines 3-20; Figs. 1, 7.

The correspondence table is retrieved using correspondence table retriever
111. See Fig. 1. If a managing ID is retrieved from the correspondence table, the file is
retrieved from memory 120 that includes correspondence table 121 and file memory
123 and output to the user. If no managing ID is obtained from the correspondence
table, an error is returned to the user. See page 22, lines 10 – 26. The file is retrieved
by “a means for retrieving files” which corresponds to at least file retriever 115. The file
retriever 0115 also sends a retrieved file to the file output unit 0103. Thus, the file
retriever 115 also corresponds to the means for sending the retrieved file to a file
outputting unit 103. File output unit 103 sends the file 2 as delivered from the data
processor 110.

The correspondence table updating part 0112 provides a function of registering the correspondence relationship of new file title, file ID and managing ID in the correspondence table 0121 or updating a registered correspondence relationship. See page 16, lines 7-12. According to the present invention, when a file ID and a file are input into the system, using file input 102, the system first checks whether the title was previously registered using “means for registering”. See p. 20, lines 2 – 10. If the file name already exists, the system checks a hash table 122 to determine whether or not the file has been previously registered. See p. 20, line 11 – 14; Fig. 5. If there is no identical hash value, which indicates identical content, the file has not been previously registered. If the hash value exists – the file to be registered is compared with the files corresponding to the managing ID obtained from the hash table 122. See Fig. 5, 506. If it is determined that a file has the same content as a previously registered file, its file management ID is set to correspond to the registered file. However, if there is no file having the same content, a new managing ID corresponding to the file is created and registered. See Fig. 5, 508; page 14, line 8-page 16, line 18, page 21, lines 13 – 22. The file is then stored with the managing ID and a correspondence table 121 is supplied by the correspondence table updating part 112 of processor 110. The managing ID corresponding to the file and file ID are finally registered for retrieval. See page 22, lines 1 – 5 and Fig. 5.

D. Claim 7

Independent claim 7 explicitly recites:

7. A file managing method for managing files, wherein a plurality of files having the same file title but different contents are managed as separate files, while also managing a plurality of files having the same content but different titles as a single file.

The present invention relates to a file management system. Particularly, the claimed filed system is able to manage files having the same title but different contents

as separate files while managing a plurality of files having different titles but the same content as a single file. See p. 1, lines 6 – 13; p. 30, lines 26 – 27.. For example, three files can be named “file A.” Assume that these three files have different contents – specifically function A, function B, and function C. Because each of these files have different contents, the files will have different Managing IDs. See p. 16, lines 19 – 25. Had two or more of these files had the same contents, the files with the same contents would share a common Managing ID. See p. 15, lines 15 – 25.

In the present file managing system a user does not have to avoid using the same file title as that of other files, as the file managing system can operate using the same file name for files having the same title as different contents. Likewise, if users give the same content different file names, the system can treat each of those as identical files – thereby eliminating the need to store multiple copies of the same file.

E. Claim 8

Independent claim 8 explicitly recites:

8. A file managing method for managing files comprising the steps of:

inputting, by a file registering person, files to be registered, the file titles thereof and a file ID;

retrieving hash tables, in which correspondence relationships of hash values of files and managing IDs are recorded, by using the hash values of the files to be registered as key values;

taking out, if a managing ID is obtained as a result of the hash table retrieval, the file corresponding to the obtained managing ID from a file memory and compares the content of the taken-out file and the contents of the files to be registered;

registering, if the content of the taken-out file is the same as the content of a file to be registered, the file title to

be registered, the file ID to be registered and the managing ID of the taken-out file in a correspondence table; and

producing, if no identity is obtained as a result of the hash table retrieval or if no same content file is detected although identity is obtained as a result of the hash table retrieval, a new managing ID, registering the new managing ID thus produced and the corresponding file to be registered in the file memory, registering the new managing ID in the hash table with the hash value of the file to be registered used as a key value, and registering the file title to be registered, the file ID to be registered and the new managing ID in the correspondence table.

The present invention relates to a file management system. According to the present invention, a file ID and a file are input into the system, using file input 102. Generally, the file request input unit 0101, the file input unit 0102, the file output unit 0103 and the data processor 0110 each have a communicating function and are interconnected via a network. File registering persons and file requesters cause the input and output of data to and from the file managing system according to the invention via the network. See p. 14, lines 19-22. The system first checks whether the title was previously registered using "means for registering". See p. 20, lines 2 – 10.

The system checks a hash table 122 to determine whether or not the file has been previously registered. See p. 20, line 11 – 14; Fig. 5. If there is no identical hash value, which indicates identical content, the file has not been previously registered. If the hash value exists – the file to be registered is compared with the files corresponding to the managing ID obtained from the hash table 122. See Fig. 5, 506.

If it is determined that a file has the same content as a previously registered file, its file management ID is set to correspond to the registered file. However, if there is no file having the same content, a new managing ID corresponding to the file is created and registered. See Fig. 5, 508; page 14, line 8-page 16, line 18, page 21, lines 13 – 22. The file is then stored with the managing ID and a correspondence table 121 is supplied by the correspondence table updating part 112 of processor 110.

The managing ID corresponding to the file and file ID are finally registered for retrieval. See page 22, lines 1 – 5 and Fig. 5. A file, file title, and file ID are input into the system using file input 102. The information is then transferred to data processor 110 when the above-described process takes place. New managing IDs are produced and recorded, if required, by data processor 110. Data processor 110 includes correspondence table retriever 111, correspondence table updating part 112, hash table retriever 113, hash table updating part 114, file retriever 115, file registering/deleting part 116, and file content comparing part 117.

In the present file managing system a user does not have to avoid using the same file title as that of other files, as the file managing system can operate using the same file name for files having the same title as different contents. Likewise, if users give the same content different file names, the system can treat each of those as identical files – thereby eliminating the need to store multiple copies of the same file. See page 32, lines 1 – 12.

F. Claim 9

Independent claim 9 explicitly recites:

9. A file managing method for managing files comprising the steps of:

inputting, by a file registering person, files to be registered, file titles thereof and a file ID;

producing new managing IDs corresponding to the files to be registered and registering the produced managing IDs and the files to be registered in a file memory;

registering file titles to be registered, a file ID to be registered and the new managing IDs in a correspondence table;

retrieving hash tables, in which correspondence relationships of hash values of files and managing IDs are

recorded, by using the hash values of the files to be registered as key values;

retrieving, when a managing ID is obtained as a result of the hash table retrieval, the file memory to take out the file corresponding to the obtained managing ID and comparing the content of the taken-out file and the contents of the files to be registered;

updating, if the content of the taken-out file is the same as a file to be registered, the new managing ID registered in the correspondence table to the managing ID corresponding to the taken-out file, and deleting the new managing ID registered in the file memory and the files to be registered from the file memory; and

registering, if no identity is obtained as a result of the hash table retrieval or if no same content file is detected although identify is obtained as a result of the hash table retrieval, the new managing ID in the hash table with the hash values of the files to be registered as key values.

The present invention relates to a file management system. According to the present invention, when a file ID and a file are input into the system, using file input 102. See p. 20, lines 2 – 10. According to the present invention, a file ID and a file are input into the system, using file input 102. Generally, the file request input unit 0101, the file input unit 0102, the file output unit 0103 and the data processor 0110 each have a communicating function and are inter-connected via a network. File registering persons and file requesters cause the input and output of data to and from the file managing system according to the invention via the network. See p. 14, lines 19-22. If there is no identical hash value, which indicates identical content, the file has not been previously registered. If there is no file having the same content, a new managing ID corresponding to the file is created and registered. See Fig. 5, 508; page 14, line 8- page 16, line 18, page 21, lines 13 – 22. The file is then stored with the managing ID and a correspondence table 121 is supplied by the correspondence table updating part 112 of processor 110. The managing ID corresponding to the file and file ID are finally registered for retrieval. See page 22, lines 1 – 5 and Fig. 5.

New managing IDs are produced and recorded, if required, by data processor 110. Data processor 110 includes correspondence table retriever 111, correspondence table updating part 112, hash table retriever 113, hash table updating part 114, file retriever 115, file registering/deleting part 116, and file content comparing part 117. See, page 18, lines 7-15.

The memory unit further includes hash tables in which relationships of hash values of files and managing IDs are recorded, the data processor includes a hash table retrieving means for retrieving the hash tables with hash values of files to be registered and a hash table registering means for registering, if no same content file has been registered in the file memory, the hash values of files to be registered and corresponding managing IDs in the hash tables, and the file content comparing means compares the content of a file corresponding to a managing ID in the case of obtaining identity as a result of the retrieval in the hash table retrieving means and the content of the pertinent file to be registered. See, pg. 9, Ins. 5-18.

The hash table retriever 0113 provides a function of retrieving the hash table 0122 by using a hash value obtained from a file as a key value to obtain a managing ID corresponding to the key value. The hash table updating part 0114, when given a file and the corresponding managing ID, calculates the key value (or hash value) in the hash table corresponding to the given file, and registers the correspondence relationship between the key value and the given managing ID in the hash table 0122. See, pg. 17, Ins. 19-27.

The file retriever 0115 provides a function of retrieving the file memory 0123 for a file corresponding to a given managing ID. The file retriever 0115 also provides a function of producing as new managing ID for a file to be registered. The file retriever 0115 further has a function of sending out a retrieved file to the file output unit 0103. The file registering/deleting part 0116 provides a function, when given a file and the corresponding managing ID, of registering the given file in the file memory 0123 with the

given managing ID or deleting a registered file from the file memory 0122 with the given managing ID. FIG. 8 shows an example of the file memory 0123. The file content comparing part 0117 compares the content of a file to be registered and the content of the files registered in the file memory 0123. See pg. 18, lns. 4-18.

The file registering/deleting part 0116 registers the file to be newly registered in the file memory 0123 with the new managing ID (step 0509). The hash table updating part 0114 registers the new managing ID in the hash table 0122 with the hash value of the newly registered file as a key value (step 0510). After the step 0511 or the step 0510, the correspondence table updating part 0112 registers the correspondence relationship of the registered file ID and the managing ID related thereto in the step 0511 or the step 0508 in the correspondence table 0121 (step 012). See pg. 21, ln. 23 et seq.

When requesting a file, a file requestor inputs a file title and file ID using file request input unit 101. The system then verifies the correspondence table 121 with a key with a given file name and file ID. See Fig. 6, 602. The correspondence table is retrieved using correspondence table retriever 111. See Fig. 1. If a managing ID is retrieved from the correspondence table, the file is retrieved from memory 120 that includes correspondence table 121 and file memory 123 and output to the user. If no managing ID is obtained from the correspondence table, an error is returned to the user. See page 22, lines 10 – 26. The file is retrieved by “a means for retrieving files” which corresponds to at least file retriever 115. The file retriever 0115 also sends a retrieved file to the file output unit 0103. Thus, the file retriever 115 also corresponds to the means for sending the retrieved file to a file outputting unit 103. File output unit 103 sends the file 2 as delivered from the data processor 110.

G. Claim 12

Independent claim 12 explicitly recites:

12. A file managing method for managing files comprising the steps of:

inputting, by a file requester, the file title of a desired file and the corresponding file ID;

retrieving a correspondence table, in which file titles, file IDs and managing IDs are recorded, with the inputted file title and file ID;

obtaining, from the correspondence table, the file title corresponding to the inputted file title and file ID and a managing ID corresponding to the inputted file ID;

retrieving a file memory, in which managing IDs and files are recorded, with the obtained managing ID;

obtaining, for the file memory, a file corresponding to the obtained managing ID; and

sending out the obtained file as the desired file to the file requester.

Referring to FIG. 6, when requesting a file, a file requestor inputs a file title and file ID using file request input unit 101. See Fig. 6, 601. According to the present invention, a file ID and a file are input into the system, using file input 102. Generally, the file request input unit 0101, the file input unit 0102, the file output unit 0103 and the data processor 0110 each have a communicating function and are inter-connected via a network. File registering persons and file requesters cause the input and output of data to and from the file managing system according to the invention via the network. See p. 14, lines 19-22. The system then verifies the correspondence table 121 with a key with a given file name and file ID. See Fig. 6, 602. The correspondence table is retrieved using correspondence table retriever 111. See Fig. 1. If a managing ID is retrieved from the correspondence table, the file is retrieved from memory 120 that includes correspondence table 121 and file memory 123 and output to the user. If no managing ID is obtained from the correspondence table, an error is returned to the user. See page 22, lines 10 – 26. If the corresponding managing ID is obtained, the file retriever 0115 takes out a file corresponding to the obtained managing ID from the file memory

013 See Fig. 6, 602. The file is retrieved by “a means for retrieving files” which corresponds to at least file retriever 115. The file retriever 0115 also sends a retrieved file to the file output unit 0103. Thus, the file retriever 115 also corresponds to the means for sending the retrieved file to a file outputting unit 103. File output unit 103 sends the file 2 as delivered from the data processor 110. See pg. 22, ln. 10 et seq.

VI. GROUNDS OF OBJECTION TO BE REVIEWED ON APPEAL

- A. The rejection of claims 1-3, 7 and 12 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,560,631 (“Ishihara”).
- B. The rejection of claims 4-6, 8-11, and 13-17 under 35 U.S.C. § 103(a) as being unpatentable over Ishihara in view of U.S. Patent No. 5,109,511 (“Nitta”).

VII. ARGUMENT

Claims 1 – 17 are pending and have been examined in the present application. All of the claims are in condition for allowance, and any pending rejections should be withdrawn.

- A. Claims 1-3, 7, and 12 are not anticipated by Ishihara

Appellants submit that claims 1-3, 7 and 12 are improperly rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,560,631 (“Ishihara”).

To anticipate a claim under 35 U.S.C. § 102, the cited reference must disclose every element of the claim, as arranged in the claim, and in sufficient detail to enable one skilled in the art to make and use the anticipated subject matter. See, PPG Industries, Inc. v. Guardian Industries Corp., 75 F.3d 1558, 1566 (Fed. Cir. 1996); C.R. Bard, Inc. v. M3 Sys., Inc., 157 F.3d 1340, 1349 (Fed. Cir. 1998). A reference that does not expressly disclose all of the elements of a claimed invention cannot anticipate

unless all of the undisclosed elements are inherently present in the reference. See, Continental Can Co. USA v. Monsanto Co., 942 F.2d 1264, 1268 (Fed. Cir. 1991).

Appellant's disagree that Ishihara discloses a file management system that discloses "managing a plurality of files having the same file title but different contents" at column 6, lines 21-26, and "managing a plurality of files having the same contents but different titles as a single file" at column 7, lines 6-13.

In Ishihara, conventional systems where every data file identified by its file name and path name in conjunction with a node name assigned to the computer that stores it as discussed. Such systems allow a plurality of computers to hold two data files sharing the same file and path names but having completely different contents. As noted in Ishihara, this is a potentially problematic situation which makes it difficult to manage the data files in a unified manner. Ishihara notes that it is desirable to introduce a unified naming convention that enables unique identification for each individual object in a distribution system environment as well as to develop a framework that provides associations between such names and their corresponding objects. Thus, there is no teaching to handle multiple files as a single object.

Computers in Ishihara store cached copies of files. These cached files are not managed by the file management system, but are local copies of remote data and program files, which are referred to as "cached files" and have the same names but different path as the original files. When a work area is created on a computer, and if the computer lacks some necessary resource files, the missing files are fetched from other computers and saved in the computer's local storage as cached files. Such cached files are stored temporarily and deleted after expiration of a predetermined period. Under the control of the process execution controller 174, cached files are accessible to authorized users in a shared manner. This shared access capability reduces the frequency of file transfer operations, thus alleviating possible increase of network traffic. Ishihara's naming convention permits each data or program file to be

uniquely identified in a distributed computing environment. Col. 7. Lns. 6-22.¹ Thus, the cached files have the same content, are uniquely identified and handled individually. In contrast, claim 1 explicitly recites a manager for managing a plurality of files having the same content but different titles as a single file.

Further, the Ishihara system allows local copies of remote data and program files which are referred to as cached files. These cached files are fetched from other computers and saved to the local storage as cached files. These cached files have the exact same title and the same content as the original as shown in Figures 11 and 12. As such, Ishihara fails to disclose a plurality of files having the same contents but different titles as asserted by the Office Action.

Appellant explicitly claims that the file managing system is capable of managing files having the same file title and different content as well as files having the same content but different titles. This is not disclosed by the single file manager in Ishihara. As such, Appellant respectfully requests reconsideration and withdrawal of this rejection.

Claims 2, 3, and 12 each require a data processor for producing and recording, if no file has the same contents as any of the files recorded in the file memory, a new managing ID for a new file to be registered in the file memory. As recited in the pending claims, when no file has the same contents as the file to be registered, a new managing ID is produced and recorded.

The Final Office Action asserts that this feature is disclosed at column 12, lines 1-16. Appellant respectfully disagrees. At column 12, lines 1-16, Ishihara discloses a process execution controller activating a warehouse server and registering various data to those servers. A resource management server then collects and records information about the warehouse servers and their respective local storage.

¹ It should be noted that FIG. 12 only shows individual cached files names, creation date, and the server the file is stored on.

Finally, a cache management server is executed. Each time a cached file is created in a warehouse server, the cache management server updates its cache management table for further management. However, at no point does Ishihara disclose producing and recording in the file memory a new managing ID for a new file if no file has the same content as any of the files to be registered.

In Ishihara, each time a cache file is created, the cache management server updates its cache management table using the same identifier as the warehouse server file. As this merely keeps the file name consistent throughout a network. Thus, Ishihara's system cannot be utilized to maintain files on a single system or use a single file management system to maintain files. In Ishihara, there are multiple file management systems each maintaining files on each of the servers. Thus, there is never a need in Ishihara for producing a new managing ID and registering in the file memory the new managing ID and a file to be registered as explicitly recited in Appellant's claim. Therefore, Appellant respectfully requests that the Examiner reconsider and withdraw the rejection to claims 2, 3, and 12.

B. Claims 4-6, 8-11, and 13-17 are not unpatentable over Ishihara in view of Nitta

Appellant submits that claims 4-6, 8-11, and 13-17 cannot be rejected under 35 U.S.C. § 103(a) as being unpatentable over Ishihara in view of U.S. Patent No. 5,109,511 ("Nitta").

Claims 4-6 and 13-17 depend from allowable claims as discussed above in Section VII A. Therefore, they too are allowable. The Final Office Action asserts that "as to claims 8-11 and 9-16, these claims recited the same subject matters as claims 2-6 and 13-15 in form of method ... hence these claims are rejected for the same reason." See Final Office Action at 5. As discussed in Section VII A above, Ishihara fails to disclose the system recited in the claims referred to by the Examiner. Nitta does

not cure the deficiency in Ishihara discussed above. Therefore all of the claims are allowable.

Appellants are not arguing this rejection separately from arguments in response to the rejection of claims 1-3, 7, and 12 above. Therefore, there are no dependant claims for which Appellants need to submit a concise explanation. See 37 C.F.R. §41.37(c)(1)(v).

C. Conclusion

For the reasons above, Appellants request the board order the withdrawal of the pending rejections.

VIII. CLAIMS

A copy of the claims involved in the present appeal is attached hereto as Appendix A. As indicated above, the claims in Appendix A include the amendments filed by Appellant on August 27, 2004.

IX. EVIDENCE

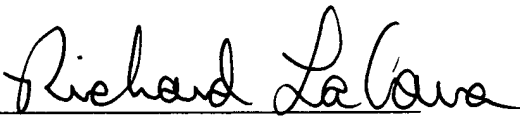
No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS

No related proceedings are referenced in II. above, or copies of decisions in related proceedings are not provided, hence no Appendix is included.

Dated: September 19, 2008

Respectfully submitted,

By 

Richard LaCava

Registration No.: 41,135
DICKSTEIN SHAPIRO LLP
1177 Avenue of the Americas
New York, New York 10036-2714
(212) 277-6500
Attorney for Appellant

APPENDIX A

Claims Involved in the Appeal of Application Serial No. 09/960,548

1. A file managing system for managing files comprising:
 - a manager for managing a plurality of files having the same file title but different contents as separate files, and for managing a plurality of files having the same content but different titles as a single file.
2. A file managing system for managing files, comprising:
 - a file input unit for sending, to a data processor, inputted files, file titles, and file IDs to be registered;
 - a file request input unit for sending, to the data processor, a file title of a requested file and a file ID;
 - a memory unit including a correspondence table and a file memory, the correspondence table including correspondence relationships of file titles, file IDs and managing IDs, the file memory for recording, managing IDs and files;
 - the data processor including:
 - a means for producing and recording, if no file has the same content as any of the files recorded in the file memory, a new managing ID for a new file to be registered in the file memory;
 - a means for retrieving files from the file memory using managing IDs;
 - a means for sending the retrieved file to a file outputting unit,

a file content comparing means for comparing the content of the new file to be registered and the contents of files registered in the file memory,

a means for registering, if a same content file has been registered in the file memory, the file title, the file ID, and the managing ID of the same content file in the correspondence table, and

a means for retrieving the correspondence table; and

the file output unit for sending, to the file request input unit, the file corresponding to the file title and the file ID requested from the file request input unit as delivered from the data processor.

3. A file managing system for managing files, comprising:

a file input unit for sending, inputted files file titles, and file IDs to be registered to a data processor;

a file request input unit for sending to the data processor, an inputted file title and a pertinent file ID;

a memory unit including;

a correspondence table, in which correspondence relationships of file titles, file IDs and managing IDs are recorded;

and a file memory, in which managing IDs and files are recorded;

a data processor including:

a means for producing a new managing ID and registering, in the file memory, the new managing ID and a file to be registered,

a file deleting means for deleting, if a same content file has been registered in the file memory, the managing ID and the file registered in the file registering means,

a means for retrieving the file memory with managing IDs for obtaining corresponding files,

a means for sending out the obtained files to a file output unit,

a file content comparing means for comparing the content of a file to be registered with the contents of the files registered in the file memory,

a correspondence table registering means for registering, in the correspondence table, the file titles to be registered, the file IDs to be registered and the new managing IDs,

a correspondence table updating means for updating, if a same content file has been registered in the file memory, the contents registered in the correspondence table registering means with the file titles to be registered, the file IDs to be registered and the managing ID of the same content file, and

a means for retrieving the correspondence table; and

the file output unit sending out, to the file request input unit, the file corresponding to the file title and the file ID requested from the file request input unit as delivered from the data processor.

4. The file managing system according to claim 2, wherein the memory unit further includes hash tables in which relationships of hash values of files and managing

IDs are recorded, the data processor includes a hash table retrieving means for retrieving the hash tables with hash values of files to be registered and a hash table registering means for registering, if no same content file has been registered in the file memory, the hash values of files to be registered and corresponding managing IDs in the hash tables, and the file content comparing means compares the content of a file corresponding to a managing ID in the case of obtaining identity as a result of the retrieval in the hash table retrieving means and the content of the pertinent file to be registered.

5. The file managing system according to claim 4, wherein the hash tables are each provided for each file title, and the hash table retrieving means decides, if no same title file as the file title of the any retrieved file has been registered in the file memory, that no hash table retrieval result is present, and retrieves, if a same title file has been registered in the file memory, the hash table corresponding to the file title of the same title file with the hash value of the pertinent file to be registered used as a key value.

6. The file managing system according to claim 4, wherein only a single hash table is provided for all file titles.

7. A file managing method for managing files, wherein a plurality of files having the same file title but different contents are managed as separate files, while

also managing a plurality of files having the same content but different titles as a single file.

8. A file managing method for managing files comprising the steps of:
 - inputting, by a file registering person, files to be registered, the file titles thereof and a file ID;
 - retrieving hash tables, in which correspondence relationships of hash values of files and managing IDs are recorded, by using the hash values of the files to be registered as key values;
 - taking out, if a managing ID is obtained as a result of the hash table retrieval, the file corresponding to the obtained managing ID from a file memory and compares the content of the taken-out file and the contents of the files to be registered;
 - registering, if the content of the taken-out file is the same as the content of a file to be registered, the file title to be registered, the file ID to be registered and the managing ID of the taken-out file in a correspondence table; and
 - producing, if no identity is obtained as a result of the hash table retrieval or if no same content file is detected although identity is obtained as a result of the hash table retrieval, a new managing ID, registering the new managing ID thus produced and the corresponding file to be registered in the file memory, registering the new managing ID in the hash table with the hash value of the file to be registered used as a key value, and registering the file title to be registered, the file ID to be registered and the new managing ID in the correspondence table.

9. A file managing method for managing files comprising the steps of:
- inputting, by a file registering person, files to be registered, file titles thereof and a file ID;
 - producing new managing IDs corresponding to the files to be registered and registering the produced managing IDs and the files to be registered in a file memory;
 - registering file titles to be registered, a file ID to be registered and the new managing IDs in a correspondence table;
 - retrieving hash tables, in which correspondence relationships of hash values of files and managing IDs are recorded, by using the hash values of the files to be registered as key values;
 - retrieving, when a managing ID is obtained as a result of the hash table retrieval, the file memory to take out the file corresponding to the obtained managing ID and comparing the content of the taken-out file and the contents of the files to be registered;
 - updating, if the content of the taken-out file is the same as a file to be registered, the new managing ID registered in the correspondence table to the managing ID corresponding to the taken-out file, and deleting the new managing ID registered in the file memory and the files to be registered from the file memory; and
 - registering, if no identity is obtained as a result of the hash table retrieval or if no same content file is detected although identify is obtained as a result of the hash

table retrieval, the new managing ID in the hash table with the hash values of the files to be registered as key values.

10. The file managing method according to claim 8, wherein the hash tables are each provided for each file title, in the hash table retrieval each hash table is retrieved for any file having the same file title as a file to be registered, decides, if no same title file has been recorded, that no retrieval result is present, and retrieves, if a same content file has been recorded, the hash table corresponding to the file title of the same title file with the hash value of the file to be registered used as a key value.

11. The file managing method according to claim 8, wherein the hash table is provided for all of file titles.

12. A file managing method for managing files comprising the steps of:
inputting, by a file requester, the file title of a desired file and the corresponding file ID;
retrieving a correspondence table, in which file titles, file IDs and managing IDs are recorded, with the inputted file title and file ID;
obtaining, from the correspondence table, the file title corresponding to the inputted file title and file ID and a managing ID corresponding to the inputted file ID;
retrieving a file memory, in which managing IDs and files are recorded, with the obtained managing ID;

obtaining, for the file memory, a file corresponding to the obtained managing ID; and

sending out the obtained file as the desired file to the file requester.

13. The file managing system according to claim 3, wherein the memory unit further includes hash tables in which relationships of hash values of files and managing IDs are recorded, the data processor includes a hash table retrieving means for retrieving the hash tables with hash values of files to be registered and a hash table registering means for registering, if no same content file has been registered in the file memory, the hash values of files to be registered and corresponding managing IDs in the hash tables, and the file content comparing means compares the content of a file corresponding to a managing ID in the case of obtaining identity as a result of the retrieval in the hash table retrieving means and the content of the pertinent file to be registered.

14. The file managing system according to claim 13, wherein the hash tables are each provided for each file title, and the hash table retrieving means decides, if no same title file as the file title of the any retrieved file has been registered in the file memory, that no hash table retrieval result is present, and retrieves, if a same title file has been registered in the file memory, the hash table corresponding to the file title of the same title file with the hash value of the pertinent file to be registered used as a key value.

15. The file managing system according to claim 13, wherein only a single hash table is provided for all file titles.

16. The file managing method according to claim 9, wherein the hash tables are each provided for each file title, in the hash table retrieval each hash table is retrieved for any file having the same file title as a file to be registered, decides, if no same title file has been recorded, that no retrieval result is present, and retrieves, if a same content file has been recorded, the hash table corresponding to the file title of the same title file with the hash value of the file to be registered used as a key value.

17. The file managing method according to claim 9, wherein the hash table is provided for all file titles.